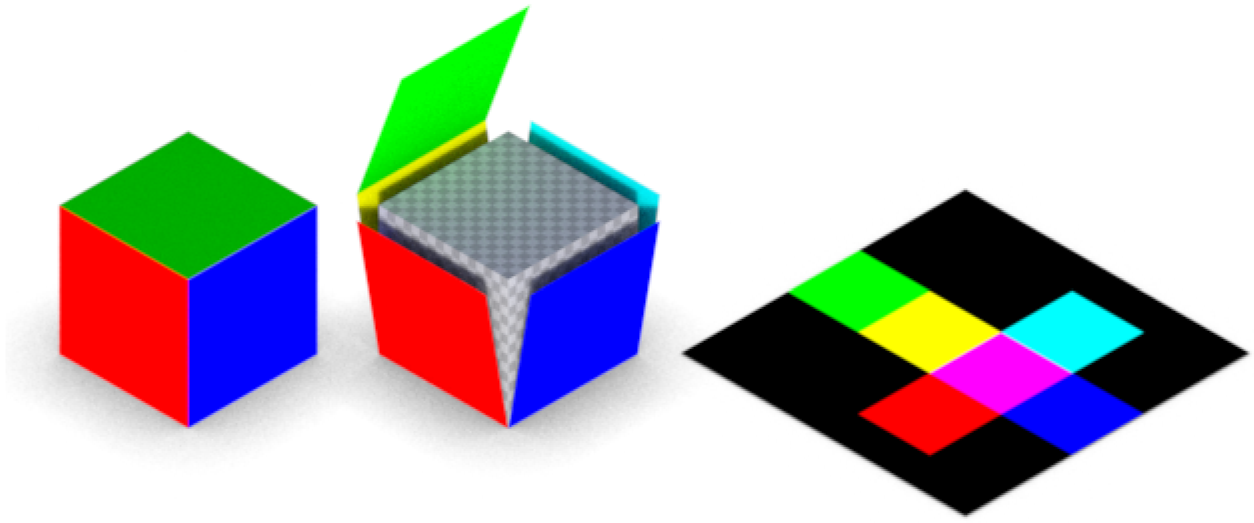
Baking Patterns



Overview

Baking allows users to pre-compute or "bake" pattern networks into 2D texture images or 3D point clouds. Baking has several advantages:

- Reduce render times be eliminating run-time computation of expensive pattern networks. These savings can be realized across many frames.
- Look development asset publishing as part of a shading pipeline.
- Lock procedural pattern information from a reference position without using Pref
- Transfer assets between renderers, real-time viewports, and 3D printing.

Baking is exposed in RenderMan via pass-through pattern plug-ins that allow users to both render and bake from the same scenes. Without heavy modification of assets or scenes, users are able to:

- Use previously baked results, if they exist, or
- Read the pattern network instead of the baked results. This can happen for any reason but typically will because the baked results do not exist or a user wishes to ignore the baked result in order to tweak the pattern network.

Baking itself is enabled by using the bake hider.

Additional details are provided in the baking app note.

Texture Baking

2D baking allows users to bake pattern signals to texture images. Users can bake directly to txmake (.tex), TIFF (.tif), and OpenEXR (.exr) outputs. 2D baking requires users to supply a well formed texture manifold, meaning there should be user created UVs laid out appropriately for their needs. While the most common texture manifold is ST, arbitrary manifolds are supported. Atlas outputs such as UDIM are supported.

RenderMan provides a generic texture baking pattern, PxrBakeTexture

⚠ Baking behavior with an overlapping texture manifold is undefined.

Point Cloud Baking

RenderMan provides a generic point cloud baking pattern, PxrBakePointCloud. This can be used to generate Ptex images.